abstractname초록 contentsname목차 listtablename표 목차 listfigurename 그림 목차

공학석사학위논문

리포머 네트워크를 이용한 음성합성 시스템

Speech Synthesis Using Reformer Network

2020년 8월

서울대학교 대학원

전기 컴퓨터 공학부

임 형 래

공학석사학위논문

리포머 네트워크를 이용한 음성합성 시스템

Speech Synthesis Using Reformer Network

2020년 8월

서울대학교 대학원

전기 컴퓨터 공학부

임 형 래

리포머 네트워크를 이용한 음성합성 시스템

Speech Synthesis Using Reformer Network

지도교수 김 남 수

이 논문을 공학석사 학위논문으로 제출함

2020년 8월

서울대학교 대학원

전기 컴퓨터 공학부

임 형 래

임형래의 공학석사 학위 논문을 인준함

2020년 8월

위 원 장: _____

부위원장: _____

위　　원: _____

요 약

Recent End-to-end text-to-speech (TTS) systems based on the deep neural network (DNN) have shown the state-of-the-art performance on the speech synthesis field. Especially, the attention-based sequence-to-sequence models have improved the quality of the alignment between the text and spectrogram successfully. Leveraging such improvement, speech synthesis using a Transformer network was reported to generate human-like speech audio. However, such sequence-to-sequence models require intensive computing power and memory during training. The attention scores are calculated over the entire key at every query sequence, which increases memory usage. To mitigate this issue, we propose the speech synthesis model based on Reformer network, the model using a Reformer network which utilizes the locality-sensitive hashing attention and the reversible residual network. As a result, we show that the Reformer network consumes almost twice smaller memory margin as the Transformer, which leads to the fast convergence of training end-to-end TTS system. We demonstrate such advantages with memory usage, objective, and subjective performance evaluation.

주요어: speech synthesis, attention-based Text-to-speech, Reformer network

학 번: 2018-23443

차 례

표 차례

그림 차례

제 1 장 INTRODUCTION

음성합성은 문자 시퀀스인 텍스트로부터 음성 신호의 형태로 변환하는 과정을 의미한다. 고전적인 음성합성은 문자 단위의 텍스트를 G2P (Grapheme-to-Phoneme)를 이용하여 음소단위로 변환한 후 발음 특성을 추출하고, 각 텍스트의 발음 길이를 추정하는 기간 모델과 음성신호의 특징을 추정하는 음향 모델을 통해 추정하게 된다. 고전적인 음성합성에서 기간 모델과 음향 모델은 HMM (Hidden Markov Model) 기반의 파라메트릭 통계 모델을 사용하였다. [1, 2] 추정된 음성 특성 시퀀스는 음성 신호처리에 기반한 보코더 (vocoder)를 통해 음성 파형 시퀀스로 변환된다. 이러한 형태로 설계된 음성합성 시스템은 여러 단계의 모델링을 필요로 하는데, 각 단계에서 중첩되는 오차가 누적되어 음성 신호의 열화가 발생할 수 있다는 단점이 존재한다. 또한 각 단계에서 언어학적 지식 및 신호처리 지식에 기반한 복잡한 설계가 요구된다는 단점 또한 존재한다.

이러한 문제를 해결하기 위해서 딥러닝 (deep learning) 기술에 기반한 종단형 (end-to-end) 음성합성 모델이 제안되었다. 딥 보이스 (Deep Voice) [3, 4], 타코트론 (Tacotron) [5, 6], 트랜스포머 TTS (Text-to-Speech) [7] 등의 모델이 제안되었고, 복잡한 설계를 요구하며 오차가 누적되는 문제를 해결하였다. 특히, 타코트론과 트랜스포머 TTS는 어텐션 메커니즘을 사용하여 텍스트와 음향 시퀀스 샘플들을 대응시켰는데, 이를 이용하면 별도의 발음 구간 정보를 주지 않더라도 학습 과정에서 두 시퀀스의 대응을 수행할 수 있다는 장점이 있다.

하지만, 어텐션 기반 음성합성 모델은 두 가지 문제를 갖고 있다. 첫 번째는 어텐션의 실패로 음성합성에서 발음 시간에 따라 어텐션 구간이 텍스트 위치에 대해 단조 증가 형태를 갖지 못하는 것이다. 이는 주로 어텐션의 에너지 값이 특정 텍스트 구간을 건너 뛰는 형태나, 특정 구간을 반복하는 방식으로 나타나며, 이를 통해 추정된 음성 또한 해당 구간을 발화하지 않거나 반복하는 것으로 나타나게 된다. 이러한

문제를 해결하는 방법으로는 단조 증가적 특징이 더 쉽게 나타날 수 있는 어텐션 함수를 사용하는 포워드 어텐션 (forward attention) [8]이나, 학습 때에 단조 증가하는 어텐션 형태를 손실 함수로 사용하는 지도 어텐션 (guided attention) [9]을 사용하는 방법 등이 제안되었다.

두 번째는 어텐션 에너지 값을 계산할 때 메모리 측면에서 비효율성을 갖는다는 문제가 있다. 트랜스포머 TTS에서는 어텐션 에너지 값을 계산할 때, 어텐션의 매 쿼리마다 키 시퀀스 전체 벡터에 대해 점 곱셈 연산을 수행한다. 그 결과, 메모리 복잡도는 쿼리의 시퀀스 길이를 L_q, 키의 시퀀스 길이를 L_k라고 했을 때, $O(L_qL_k)$의 값을 갖게 된다. 게다가 트랜스포머 TTS에는 인코더의 재귀 어텐션, 디코더의 재귀 어텐션과 인코더-디코더 어텐션이 각각 3번씩 중첩된 형태를 갖고 있어 학습 시 큰 규모의 메모리 사용을 필요로 한다. 하지만, 음성합성에서 배치 사이즈는 합성음의 음질을 높이는 데에 중요한 요소이며, 트랜스포머 TTS에서는 배치 사이즈가 안정적인 학습에 중요한 역할을 한다고 보고된 바 있다 [7]. 따라서, 메모리를 효율적으로 사용하여 어텐션 기반 모델을 학습할 수 있다면 더 안정적인 학습과, 음질이 보다 좋은 합성음을 생성할 수 있을 것이다.

학습된 트랜스포머 구조에서 어텐션 에너지 값을 계산할 때 하나 혹은 두 개의 키 벡터를 제외한 거의 모든 키 벡터에 대해서 어텐션 에너지 값이 0에 가깝게 나오는 것을 확인할 수 있다. 그러므로 어텐션 에너지 값이 0으로 계산되는 모든 키 벡터들에 대해서 어텐션 연산을 수행하는 것은 비효율적이다. 최근 제안된 리포머 네트워크 [10]는 위치-민감성 해시 어텐션 [11]을 사용하여 어텐션 연산 과정에서 메모리 측면에서 효율적인 계산을 수행한다. 또한, 가역 잔여 네트워크 (reversible residual network) [12]를 사용하여 잔여 네트워크 [13]의 중첩시 신경망의 활성 함수 결과를 저장해야하는 특성을 제거하여 메모리 측면에서의 효율성을 강화하였다. 두 방식을 사용하여 구성한 리포머 네트워크는 자연어 이해 분야에서 성능의 큰 저하 없이 메모리 사용을 낮춰 효율적인 학습이 가능함이 보고되었다.

본 논문에서는 리포머 네트워크를 이용한 음성합성 시스템을 제안한다. 어텐션

모듈 중 메모리 소비가 큰 디코더의 재귀 어텐션에 위치-민감성 해싱 어텐션과 가역 잔여 네트워크를 사용하여 효율적으로 메모리를 사용하여 학습을 수행하였다. 또한, 효과적인 학습을 위해 포워드 어텐션을 차용하여 인코더-디코더 어텐션에서 단조증가적인 형태를 보다 쉽게 형성할 수 있도록 하였다. 본 논문의 구성은 하기와 같이 구성되었다.

먼저 2장에서는 트랜스포머 기반 음성합성 시스템과 리포머 네트워크, 그리고 포워드 어텐션 기법에 대한 설명을 기술하였다. 3장에서는 본 논문을 통해 제안하는 리포머 네트워크를 사용한 음성합성 시스템에 대하여 기술하며, 4장에서는 실험 환경 및 실험 과정과 결과에 대한 분석 내용을 기술하였다. 마지막으로 5장에서는 결론과 실험에 대한 논의를 기술하였다. 실험에 사용한 음성 특징은 275 샘플을 1 프레임으로 나타내는 프레임 단위의 멜 스펙트로그램을 사용하였으며, 멜 스펙트로그램에서 WaveGlow 보코더 [14]를 이용하여 음성을 복원하였다.

제 2 장 Background

2.1 Transformer TTS

2.1.1 Feature extraction

트랜스포머 기반 음성합성 시스템에서는 텍스트와 음성신호를 각각 인코더와 디코더에 입력하기 전에 특징 추출 과정을 거치게 된다. 이를 전처리신경망 (PreNet)이라고 하며, 전처리 신경망은 컨벌루션 신경망과 선형 신경망으로 구성되어 있으며 전처리 신경망의 출력에 위치 임베딩 시퀀스를 더한다. 전처리 신경망에서는 컨벌루션 신경망과 선형 신경망을 통해서 텍스트로부터 음향 특징 추정에 필요한 특징을 신경망 학습을 통해 별도의 언어학적 지식 없어도 추출할 수 있게 된다. 1보다 큰 커널 사이즈의 컨벌루션 신경망을 이용하면 시퀀스의 특정 시점 뿐만 아니라 전후 샘플들을 반영하여 맥락 정보가 반영된 특징을 추출할 수 있다. 음성 스펙트로그램은 실제 생성시 순차적으로 생성되므로 학습 시에만 사용 가능하며, 실제 생성시에는 사용할 수 없기 때문에 텍스트를 입력으로 받는 전처리 신경망에서만 1보다 큰 커널 사이즈를 갖는 컨벌루션 신경망을 사용한다. 음성 스펙트로그램에서는 문맥 정보를 반영하지 않고 샘플 단위로 분석하는 피드 포워드 네트워크를 사용한다. 위치 임베딩은 트랜스포머 네트워크를 학습시킬 때 재귀 어텐션 계산 시, 시퀀스 내 각 벡터 값에만 영향을 받고 시퀀스의 순서에 독립적으로 계산되는 문제를 해결하기 위해 추가된다. 위치 임베딩은 다양한 진동수의 사인 함수 값의 시퀀스를 사용하여 생성하며, 신경망을 통과한 텍스트와 음성 특징 시퀀스에 더해진다.

전처리 신경망을 통과한 텍스트 시퀀스와 음성 시퀀스는 다음과 같이 나타낼 수 있다.

$$\boldsymbol{x'}_{1:N} = PreNet_{enc}(\boldsymbol{x_{1:N}}) + \epsilon \times PE(1:N) \tag{2.1}$$

$$\boldsymbol{y'}_{1:T} = PreNet_{dec}(\boldsymbol{y}_{1:T}) + \epsilon \times PE(1:N) \tag{2.2}$$

$$PE(pos) = cat_{i=1:h}(PE_i(pos)) \tag{2.3}$$

$$PE_i(pos) = \begin{cases} sin(\frac{pos}{10000^{\frac{2i}{h}}}), & \text{if } i \text{ is even} \\ cos(\frac{pos}{10000^{\frac{2i}{h}}}), & \text{if } i \text{ is odd} \end{cases} \tag{2.4}$$

$x_{1:N}$은 텍스트 시퀀스, $y_{1:T}$는 멜 스펙트로그램 시퀀스를 나타내며 PE는 위치 임베딩을 나타낸다. PE는 h 차원의 삼각함수 값의 연쇄로 나타내며 ϵ는 위치 임베딩의 가중치, h는 전처리 신경망 출력의 차원을 나타낸다. ϵ 값을 크게 설정할 경우 재귀 어텐션에서 위치 정보를 더 강조해서 입력할 수 있다. 전처리 신경망과 위치 임베딩을 이용하여 인코더와 디코더의 입력될 텍스트와 멜 스펙트로그램의 특징 벡터를 추출한다.

2.1.2 Attention based encoder and decoder

재귀 어텐션을 사용하면 반복 신경망 (Recurrent Neural Network)에서 장기 의존성이 사라지는 문제를 해결할 수 있다. 또한, 반복 신경망에서는 시퀀스 샘플에 대해 순차적으로 학습하는 반면, 재귀 어텐션을 이용하면 시퀀스 전체 샘플에 대해 병렬적으로 계산이 가능하므로 학습 속도가 느린 문제를 해결할 수 있다. 인코더와 디코더는 재귀 어텐션과 선형 신경망의 잔여 네트워크로 구성되며 이를 반복하는 형태로 표현력을 증가시킨다. 잔여 네트워크는 신경망 학습 과정에서 경사 완화(vanishing gradient) 문제를 해결하여 입력의 특성을 결과에 잘 반영시킨다는 장점이 있다. 인코더와 디코더의 매 블록에서 잔여 네트워크를 사용함으로써 복잡한 구성의 트랜스포머 모델에서 입력 특성을 활용할 수 있다. 트랜스포머 음성합성의 인코더와 디코더의 각 모듈은 다음 식과 같이 나타낼 수 있다.

$$\boldsymbol{\alpha}^{E}_{1:N} = Attn(q_{enc}(\boldsymbol{x'}_{1:N}), k_{enc}(\boldsymbol{x'}_{1:N})) \tag{2.5}$$

$$\boldsymbol{\alpha}^{D}_{1:T} = Attn(q_{dec}(\boldsymbol{y'}_{1:T}), k_{dec}(\boldsymbol{y'}_{1:T})) \tag{2.6}$$

$$\boldsymbol{h}^{x}_{n} = FFN_{enc}(\sum_{i}^{N}(\boldsymbol{\alpha}^{E}_{n}(i) v_{enc}(\boldsymbol{h}^{x}_{i}))) \tag{2.7}$$

$$\boldsymbol{h}^{y}_{t} = FFN_{dec}(\sum_{j}^{T}(\boldsymbol{\alpha}^{D}_{t}(j) v_{dec}(\boldsymbol{h}^{y}_{j}))), \tag{2.8}$$

$\boldsymbol{\alpha}$는 어텐션 가중치 값이고, $Attn$은 멀티헤드 어텐션이다. q, k, v는 어텐션 메커니즘 적용시 필요한 가중치 매트릭스로 각각 쿼리, 키, 밸류 매트릭스이며, FFN은 선형 신경망으로 구성된 네트워크이다. 인코더와 디코더의 재귀 어텐션을 수행하면서 텍스트와 멜 스펙트로그램의 특징 시퀀스는 장기 의존성을 가지게 되며 다음 중첩의 입력으로 사용된다.

재귀 어텐션의 결과로 얻은 텍스트와 멜 스펙트로그램의 특징 벡터는 인코더-디코더 어텐션에서 정렬된다. 멜 스펙트로그램의 특징 벡터가 쿼리의 입력으로, 텍스트의 특징 벡터가 키와 밸류의 입력으로 사용된다. 쿼리 시퀀스와 키 시퀀스의 점 연산으로 어텐션 가중치를 구하여 밸류 시퀀스의 선형 합을 어텐션 출력으로 내보낸다. 멜 스펙트로그램 샘플 별로 관련이 높은 텍스트가 키 벡터와의 점연산을 통해 다음 멜 스펙트로그램 샘플을 효과적으로 추정할 수 있도록 가중치가 결정된다. 이 가중치를 이용하여 계산된 텍스트의 밸류 시퀀스 선형 합이 선형 신경망과 잔여 네트워크를 통과하여 멜 스펙트로그램의 추정 입력으로 사용하게 된다. 인코더-디코더 네트워크는 다음 식으로 나타낼 수 있다.

$$\boldsymbol{\alpha}_{1:N} = Attn(q(\boldsymbol{h}^{y}_{1:T}), k(\boldsymbol{h}^{1:N})) \tag{2.9}$$

$$\boldsymbol{h} = \sum_{i}^{N} (\boldsymbol{\alpha}_{n}(i) v_{enc}(\boldsymbol{h}_{i})) \tag{2.10}$$

$$\boldsymbol{c} = FFN(\boldsymbol{h}) + \boldsymbol{h} \tag{2.11}$$

$\boldsymbol{\alpha}_{1:N}$는 멜 스펙트로그램에 대한 텍스트 시퀀스의 가중치이며 $\boldsymbol{h}$는 가중치에 대해 텍스트 정보의 밸류 값을 선형 합한 것이다. $\boldsymbol{c}$는 선형 신경망과 잔여 네트워크를 통과하여 다음 단계의 멜 스펙트로그램의 추정에 대한 입력이 된다. 인코더와 디코더 네트워크는 각각 위의 수식으로 구성된 블록이 3번 중첩되는 형태를 갖는다. 블록을 중첩함으로써 음성합성 모델의 표현력을 증가시킬 수 있으며, 인코더-디코더 어텐션 네트워크에서 특정 블록에서 텍스트와 스펙트로그램간의 정렬이 누락될 경우 다른 블록에서 연결되는 경우도 있어 안정적으로 텍스트와 스펙트로그램간의 정렬이 가능한 효과가 있다.

2.1.3 Postnet

텍스트와 멜 스펙트로그램으로부터 어텐션 기반의 인코더와 디코더를 거쳐 생성된 출력으로부터 다음 시간 단계의 멜 스펙트로그램과 스탑 토큰을 추정하게 된다. 다음 시간 단계의 멜 스펙트로그램을 추정할 때, 디코더의 출력을 입력으로 받아 선형 신경망과 컨벌루션 신경망의 잔여 네트워크로 추정한다. 스탑 토큰은 한 층의 선형 신경망을 통과하여 추정하는데, 이는 어느 멜 시간 단계에서 주어진 발화가 종료되는지 추정하기 위한 출력이다.

$$\boldsymbol{c'} = Linear_{mel}(\boldsymbol{c}) \tag{2.12}$$

$$\hat{\boldsymbol{y}} = Conv(\boldsymbol{c}) + \boldsymbol{c'} \tag{2.13}$$

$$pred_{stop} = Linear_{stop}(\boldsymbol{c}) \tag{2.14}$$

2.1.4 Loss function

트랜스포머 기반 음성합성에서 추정된 멜 스펙트로그램과 스탑 토큰을 이용하여 손실 함수를 설계한다. 추정된 멜 스펙트로그램은 L1 손실함수를 사용하고, 스탑 토큰은 이진 크로스 엔트로피를 사용한다. 추정된 멜 스펙트로그램과 정답 멜 스펙트로그램의 차를 줄여 점점 정확한 멜 스펙트로그램을 추정할 수 있도록 하고, 해당 시간 단계에서 추정된 디코더 출력으로부터 추정된 스탑토큰에서는 발화가 끝난 시점인지에 대한 라벨을 이진 크로스 엔트로피를 이용하여 추정한다. 이 때 사용하는 스탑 토큰 라벨은 정답 멜 스펙트로그램의 첫 샘플부터 마지막 시간의 직전까지 0으로 설정하며, 마지막 시간 단계에서와 배치 단위 학습 시 가장 긴 멜 스펙트로그램 시퀀스의 길이에 맞춰 0-패딩 하는 시간 범위에서 1이 되도록 한다. 이진 크로스 엔트로피 손실 함수를 포함한 분류 작업의 손실 함수에서 라벨 간 빈도의 불균형이 안정적인 학습에 방해가 되는 문제가 있는데, 이를 보완하기 위해 스탑 토큰 추정은 손실 함수에서 스탑 토큰이 1이 되는 경우의 가중치를 증가시켜 학습하였다. 스탑 토큰을 이용하여 학습 후 실제 음성 생성 시 매 프레임에서 스탑 토큰을 추정하여 스탑 토큰 확률이 일정치를 넘으면 생성을 중단하는 방식으로 음성 생성 알고리즘을 구성할 수 있다.

2.2 Reformer network

트랜스포머 네트워크는 어텐션에 기반한 네트워크로 어텐션은 두 시퀀스간의 상관관계를 쿼리와 키 시퀀스의 모든 스텝에 대해서 계산하게 된다. 따라서 어텐션 모듈마다 쿼리의 시퀀스 길이를 L_q, 키의 시퀀스 길이를 L_k라고 했을 때, 메모리 복잡도는 $O(L_qL_k)$의 값을 갖게 된다. 하지만 훈련된 트랜스포머 네트워크의 어텐션은 전체 시퀀스 중 일부를 제외하면 어텐션 가중치 값이 0에 가까운 것을 확인할 수 있다. 시퀀스 중 가중치 값이 0이 되는 샘플들에 대해 어텐션 계산을 하지 않으면 비효율적인 메모리 소모를 막을 수 있다. 또한, 잔여 네트워크를 사용할 때 신경망의 역전파 과정에서 그래디언트를 계산할 때 해당 잔여 레이어의 활성화 값을 갖고 있어야 한다는 단점이 있는데, 트랜스포머 모델은 잔여 네트워크가 연속적으로 반복되는 형태이기 때문에 매 레이어에서 잔여 레이어의 활성화 값을 메모리에 저장해두어야 한다. 잔여 레이어의 활성화 값을 저장하지 않고 알 수 있다면 메모리를 절약하여 신경망 학습을 진행할 수 있다.

2.2.1 Locality-sensitive hashing attention

리포머 네트워크는 위치-민감성 해싱 어텐션을 사용하여 어텐션 가중치가 높을 가능성이 있는 키 벡터들의 군집화를 통해 비효율적인 메모리 소모를 완화한다. 일반적으로 해시는 해시 키 값에 독립적으로 해시 출력이 결정되지만, 서로 거리가 가까운 벡터가 같은 해시 출력을 갖는 위치-민감성 해싱을 사용하여 쿼리와 키의 거리가 가까운 것들끼리 군집화한다. 군집화된 쿼리, 키 벡터들 안에서 어텐션을 수행하면 적은 메모리 소모로도 어텐션 메커니즘을 적용할 수 있다.

그림 2.1은 위치-민감성 해싱 어텐션을 사용하는 과정을 나타낸 것이다. 먼저 쿼리 시퀀스와 키 시퀀스를 결합시킨 후, 위치-민감성 해시 함수를 이용하여 시퀀스 내의 샘플들을 매핑한다. 그림 2.1의 오른쪽은 위치-민감성 해시 함수의 예시로,

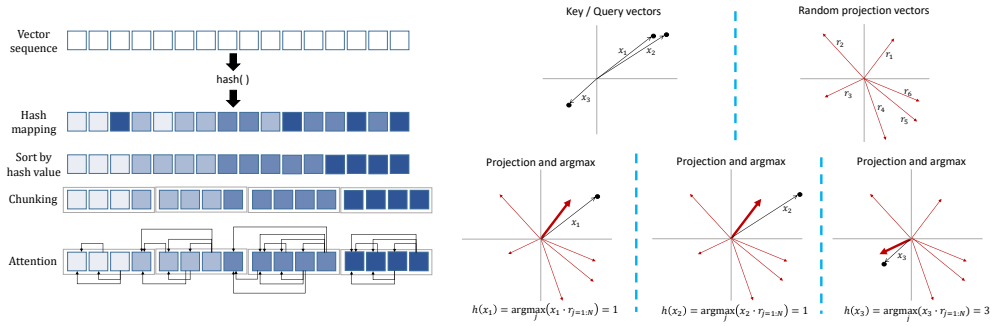

그림 2.1: Procedure of locality-sensitive hashing attention.

일정 크기 내에서 무작위로 선언한 투사 벡터의 집합을 $\boldsymbol{R}$라 하면 해시 함수 h는

$$h(\boldsymbol{x}) = argmax(\boldsymbol{x}\boldsymbol{R}) \tag{2.15}$$

로 나타낼 수 있다. 해시 버킷 사이즈는 투사 벡터의 개수로 그림에서는 6개이다. 그림 내 x_1과 x_2 벡터는 거리가 가깝기 때문에 6개의 투사 벡터 중 같은 투사 벡터에 대해 투사한 값이 가장 클 가능성이 높다. 반면, x_3은 두 벡터에 비해 거리가 멀기 때문에 다른 투사 벡터에 투사한 값이 최대일 가능성이 높다. 위와 같은 과정을 거쳐 해시 출력을 시퀀스에 대해 계산하고 이를 같은 해시 출력을 갖는 샘플에 대해 버켓팅을 수행한다. 그 후, 정렬된 시퀀스를 일정 크기로 청킹한 후, 자기 자신의 청크와 인접한 한 청크에 대하여 어텐션 연산을 수행한다. 이 과정에서 같은 청크에 있더라도 해시 출력이 다른 샘플끼리는 마스킹을 통해 어텐션 계산을 수행하지 않도록 한다. 위치-민감성 해싱 어텐션을 사용하여 어텐션 메커니즘을 수행하는 과정은 다음과 같다.

$$\boldsymbol{\alpha}_i[j] = softmax(\sum_{i \in P} \frac{exp(\boldsymbol{q}_i \cdot \boldsymbol{k}_j - m(j, \boldsymbol{P_i}))}{d_{model}}) \tag{2.16}$$

$$m(x, \boldsymbol{A}) = \begin{cases} 0, & \text{if } x \in \boldsymbol{A} \\ \infty, & \text{otherwise} \end{cases} \tag{2.17}$$

$$\boldsymbol{P_i} = \{j | h(\boldsymbol{q}_i) = h(\boldsymbol{k}_j)\} \tag{2.18}$$

$$\boldsymbol{c}_t = \sum_{i}^{N} (\boldsymbol{\alpha}_t[j] \boldsymbol{h}_j) \tag{2.19}$$

α는 어텐션 가중치이며, $m()$은 마스킹 함수이다. 마스킹 함수는 $\boldsymbol{P_i}$에 속하지 않는 쿼리와 키 샘플에 대해 어텐션 에너지 값이 0이 되도록 하며, $\boldsymbol{P_i}$는 현재 쿼리 샘플을 입력으로 받는 해시 출력이 같은 샘플 인덱스의 집합이다. 이 과정을 통해 같은 해시 출력을 갖는 쿼리, 키 샘플끼리 어텐션 에너지 값을 계산하도록 하여 메모리 소모를 줄일 수 있다.

2.2.2 Reversible residual network

잔여 네트워크는 경사 완화 (vanishing gradient)를 해소할 수 있는 좋은 수단으로써 신경망 네트워크의 여러 분야에서 우수한 성능을 보인다. 학습 시 역전파 과정에서 잔여 네트워크 출발 지점의 신경망 활성화 값을 저장해두어야 경사도 계산이 가능하지만, 가역 잔여 네트워크는 역전파 과정에서 해당 위치의 신경망 활성화 값을 순차적으로 계산해 나가면서 접근이 가능하게 된다.

그림 2.2은 잔여 네트워크와 가역 잔여 네트워크의 구조이다. 그림 2.2의 오른쪽 구조를 사용하면 역전파 과정에서 잔여 네트워크 출발점의 신경망 활성화 값을 잔여 네트워크의 합 지점의 신경망 활성화 값을 이용하여 구할 수 있다. 이를 통해 블럭이 중첩된 구조인 인코더와 디코더에서 반복적으로 구성되는 잔여 네트워크에서

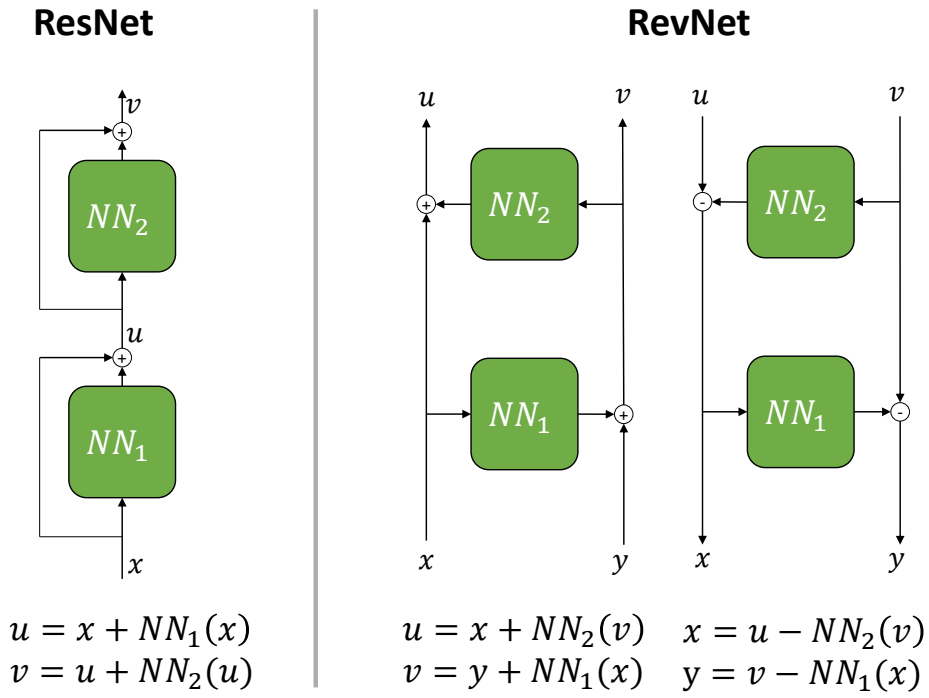

그림 2.2: Residual network (left) and reversible residual network(right).

역전파시 매 잔여 네트워크마다 신경망 활성화 값이 저장되는 비효율성을 해소할 수 있다.

2.3 Forward attention

음성합성에서 발화할 텍스트는 대부분의 언어에서 순차적으로 나열된다. 포워드 어텐션은 이를 반영하여 매 쿼리 샘플에서의 어텐션 가중치 중 비율이 높은 키 샘플이 이전 쿼리 샘플의 키 샘플과 같거나 키 시퀀스 상의 다음 샘플에 가중치가 높도록 구성된 알고리즘이다. 포워드 어텐션은 두 가지 알고리즘(바닐라 포워드 알고리즘과 전이 에이전트 포워드 알고리즘)이 있다. 바닐라 포워드 알고리즘은 일반적인 어텐션 가중치에 새로운 가중치 값이 곱해지는 형태로 구해지며, 새로운 가중치 값은 키 시퀀스의 순서상 연속된 값들에 일반적인 어텐션 가중치가 곱해져서 순차적인 어텐션 정렬이 잘 형성되도록 유도한다. 또한, 전이 에이전트 포워드 알고리즘은 새로운 가중치 값에서 연속된 두 값에 새로운 가중치 파라미터를 이용하여 비율을 결정하게 된다. 각각의 알고리즘의 어텐션 가중치 계산 과정은 아래 표 2.1, 2.2와 같다.

Algorithm 2 Forward attention

Initialize:

$\boldsymbol{\alpha}_0(1) \leftarrow 1.0$

$\boldsymbol{\alpha}_0(n) \leftarrow 0.0, n = 2, ..., N$

for t=1 to T do:

$\boldsymbol{y}_t(n) \leftarrow Attend(\boldsymbol{x}, \boldsymbol{q}_t)$

$\boldsymbol{\alpha}'_t(n) \leftarrow (\boldsymbol{\alpha}_{t-1}(n) + \boldsymbol{\alpha}_{t-1}(n-1))\boldsymbol{y}_t(n)$

$\boldsymbol{\alpha}_t(n) \leftarrow \boldsymbol{\alpha}'_t(n) / \sum_{i=1:N} \boldsymbol{\alpha}'_t(i)$

$\boldsymbol{c}_t \leftarrow \sum_{i=1:N} \boldsymbol{\alpha}_t(i)\boldsymbol{x}_i$

end for

표 2.1: Forward attention

Algorithm 3 Forward attention with transition agent

Initialize:

$\boldsymbol{\alpha}_0(1) \leftarrow 1.0$

$\boldsymbol{\alpha}_0(n) \leftarrow 0.0, n = 2, ..., N$

$\boldsymbol{u}_0 \leftarrow 0.5$

for t=1 to T do:

$\boldsymbol{y}_t(n) \leftarrow Attend(\boldsymbol{x}, \boldsymbol{q}_t)$

$\boldsymbol{\alpha}'_t(n) \leftarrow ((1 - \boldsymbol{u}_{t-1})\boldsymbol{\alpha}_{t-1}(n) + \boldsymbol{u}_{t-1}\boldsymbol{\alpha}_{t-1}(n-1))\boldsymbol{y}_t(n)$

$\boldsymbol{\alpha}_t(n) \leftarrow \boldsymbol{\alpha}'_t(n) / \sum_{i=1:N} \boldsymbol{\alpha}'_t(i)$

$\boldsymbol{c}_t \leftarrow \sum_{i=1:N} \boldsymbol{\alpha}_t(i)\boldsymbol{x}_i$

$\boldsymbol{u}_t \leftarrow DNN(\boldsymbol{c}_t, \boldsymbol{o}_{t-1}, \boldsymbol{q}_t)$

end for

표 2.2: Forward attention with transition agent

위와 같은 포워드 어텐션 알고리즘을 사용할 경우 일반적인 어텐션 알고리즘과 비해 학습 과정에서 보다 빨리 텍스트와 스펙트로그램간의 정렬이 이루어질 수 있다. 한편, 위치-민감성 해싱 어텐션의 과정에서 학습 초기에 모든 키 샘플에 대해 어텐션 가중치 값을 계산하지 않기 때문에 학습이 불안정할 수 있는데, 포워드 어텐션 알고리즘을 사용하면 위치-민감성 해싱 어텐션의 학습 초기 불안정성을 보완하여 보다 안정적인 학습이 가능하다. 또한, 학습이 진행된 이후에는 음성합성 시 문자의 생략이나 반복을 줄여 생성된 발화의 이해도 및 명료도를 높여 합성된 음성의 음질을 높일 수 있다.

제 3 장 Proposed method

3.1 Memory efficient Reformer TTS

본 논문에서는 리포머 네트워크를 구성하는 위치-민감성 해싱 어텐션과 가역 잔여 네트워크를 활용하여 학습 과정에서 메모리를 효율적으로 사용하는 리포머 네트워크 기반 음성 합성 시스템에 대해 다룬다. 트랜스포머 기반 음성합성은 반복 신경망의 장기 의존성 문제를 해결하고 빠른 학습 속도를 보이지만, 학습 과정에서 메모리의 비효율적인 단점이 있다. 이러한 단점을 리포머 네트워크를 이용하여 해소하고, 리포머 네트워크의 위치-민감성 해싱 어텐션이 모든 키 시퀀스에 대해 어텐션 계산을 하지 않는다는 점을 포워드 어텐션을 사용하여 보완하였다. 본 논문에서 제안하는 모델의 전체 구조는 그림 3.1과 같다.

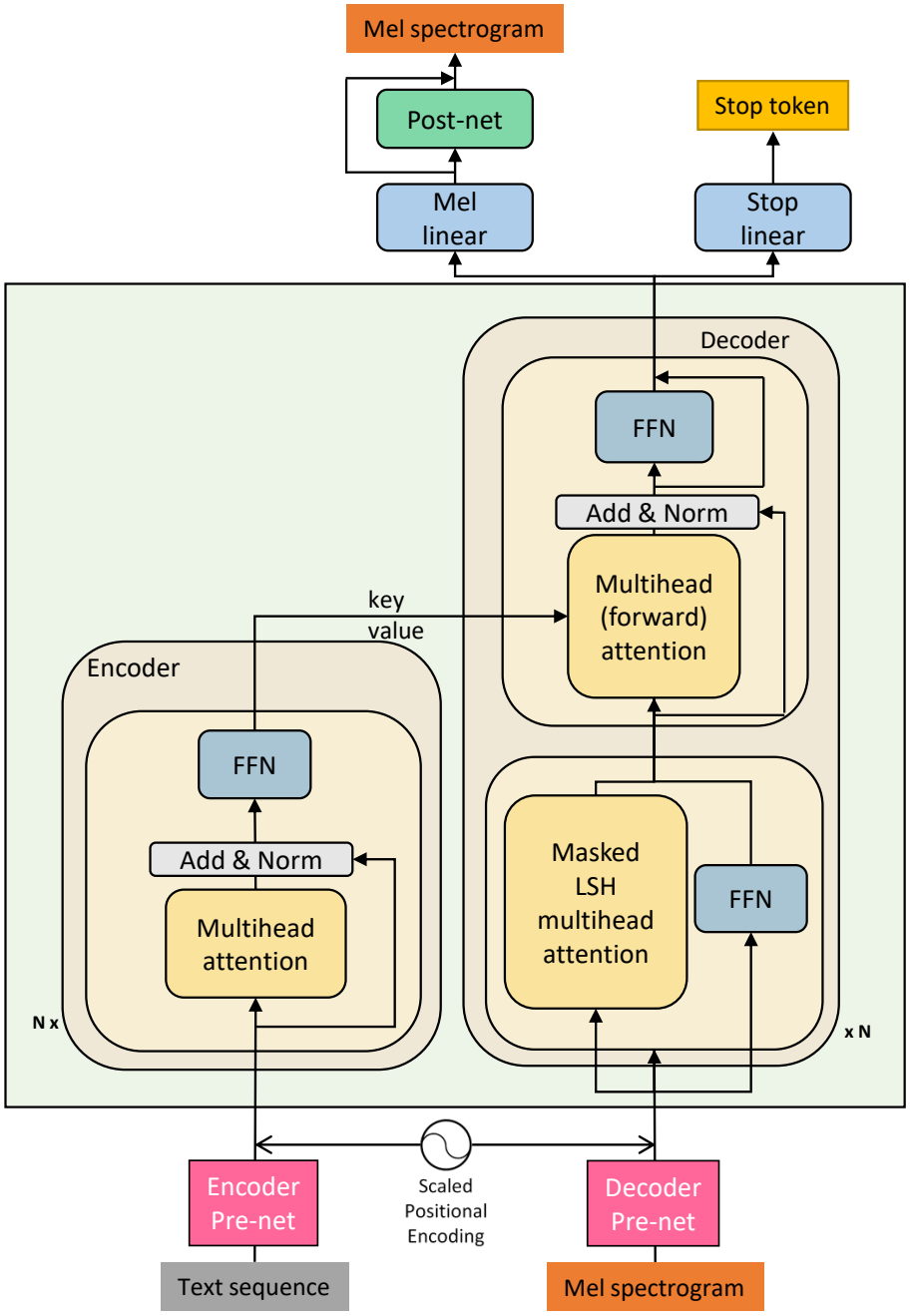

그림 3.1: Overall architecture of the proposed model.

3.1.1 feature extraction

텍스트의 특징 추출의 경우 먼저 임베딩 테이블을 통해 텍스트 시퀀스로부터 512차 임베딩 벡터를 추출한다. 5의 커널 크기를 갖는 3개의 연속된 256차 컨벌루션 신경망을 거쳐 텍스트의 특징 벡터를 추출하게 된다. 매 컨벌루션 계산시 배치 정규화(batch normalization)와 0.2의 dropout을 수행하여 안정적으로 학습이 진행되도록 하며, 활성 함수로는 ReLU (Rectified Linear Unit)를 사용하였다. 컨벌루션 신경망 출력 후에도 시퀀스 길이를 유지하기 위해 매 컨벌루션 연산 시 0-패딩을 수행하여, 256차의 컨텍스트 시퀀스를 출력으로 얻게 된다. 한편, ReLU의 출력의 범위가 [0, + ∞)이므로 위치 임베딩과 전처리 신경망의 중간값을 일치시키기 위해 선형 신경망을 추가하였다. 마지막으로 위치 정보를 추가하기 위하여 트랜스포머 기반 음성합성에서 사용하는 위치 임베딩 시퀀스를 더하여 인코더에 입력하게 된다. 멜 스펙트로그램의 특징을 추출할 때는 2개의 연속된 선형 신경망을 사용하였다. 선형 신경망의 활성 함수로는 ReLU를 사용하였고, 0.5의 dropout을 적용하였다. 텍스트의 특징 추출과 마찬가지로 위치 임베딩과 전처리 신경망의 중간값을 일치시키기 위해 선형 신경망을 추가하였다. 전처리 신경망의 출력에 위치 임베딩 시퀀스를 더하여 디코더의 재귀 어텐션에 입력된다.

3.1.2 Encoder

인코더에서 메모리를 효율적으로 사용하기 위해 위치-민감성 해싱 어텐션을 사용할 경우, 컨텍스트 시퀀스를 효과적으로 추정하지 못하는 것을 확인하였다. 텍스트는 멜 스펙트로그램 시퀀스에 비해 길이가 짧은데 투사 벡터의 개수인 버킷 크기를 작게 설정할 경우, 쿼리 샘플이 키 시퀀스의 샘플과 대응하기 어려워진다. 반대로 버킷 크기를 크게 설정할 경우 전체 시퀀스에 대해 어텐션 값을 계산하는 것과 메모리 측면에서 차이가 없게 된다. 시퀀스의 길이가 디코더의 입력에 비해 작은 인코더에서는 256차의 멀티 헤드 재귀 어텐션과 잔여 네트워크로 구성하였으며, 출력을 디코더의 인코더-디코더 어텐션의 입력으로 사용하게 된다.

3.1.3 Decoder

디코더에서는 특징이 추출된 멜 스펙트로그램과 인코더를 거친 텍스트의 컨텍스트 시퀀스를 이용하여 다음 시간 스텝의 멜 스펙트로그램을 추정하게 된다. 버킷 크기 32, 256차의 위치-민감성 해싱 어텐션이 디코더의 재귀 어텐션에 사용되며, 가역 잔여 네트워크를 거쳐 인코더-디코더 어텐션의 입력으로 사용된다. 위치-민감성 해싱 어텐션은 인코더-디코더 어텐션에서는 멜 스펙트로그램과 텍스트의 의미차로 인해 사용하지 않았다. 위치-민감성 해싱 어텐션은 쿼리 시퀀스와 키 시퀀스의 거리가 가까운 점을 이용하게 되는데, 재귀 어텐션과 달리 쿼리 시퀀스와 키 시퀀스가 서로 다른 종류의 정보를 갖고 있기 때문에, 벡터 상에서 거리가 가깝다고 하더라도 두 벡터는 연관이 없을 가능성이 크기 때문이다. 인코더-디코더에서는 256차의 멀티 헤드 어텐션을 사용하였고, 그 후 선형 신경망과 잔여 네트워크를 통해 멜 스펙트로그램과 스탑 토큰을 추정하는 출력을 내보내게 된다. 한편, 위치-민감성 해싱 어텐션은 쿼리 샘플에 대해 키 시퀀스의 일부에 대해 어텐션 계산을 수행하기 때문에 트랜스포머 네트워크의 재귀 어텐션에 비해 다소 불안정하다. 또한, 음성합성의 특성상 모든 텍스트 시퀀스에 대해 순차적으로 반복이나 생략 없이 단조증가 형태의 어텐션을 형성해야 하는데, 이를 보완하기 위해 어텐션의 단조증가 특성이 쉽게 나타나도록 하는 포워드 어텐션 알고리즘을 사용하였다. 포워드 어텐션 알고리즘 중 전이 에이전트 (transition agent) 형태의 알고리즘을 사용하였고, 이를 적용하였을 때, 반복이나 생략이 발생하는 점을 일부 해소하였다. 멀티 헤드 어텐션에 사용하였으며, 멀티 헤드 포워드 어텐션 수식 및 알고리즘은 다음과 같다.

$$head^{\boldsymbol{Q}}_{1:H} = Chunk(\boldsymbol{Q}, H) \tag{3.1}$$

$$head^{\boldsymbol{K}}_{1:H} = Chunk(\boldsymbol{K}, H) \tag{3.2}$$

$$head^{\boldsymbol{V}}_{1:H} = Chunk(\boldsymbol{V}, H) \tag{3.3}$$

$$Attn_{1:H} = softmax(\frac{\boldsymbol{Q}_{i:H}\boldsymbol{K}^T_{i:H}}{\sqrt{d_k}}) \tag{3.4}$$

Algorithm 3 Multihead forward attention with transition agent

Initialize:

$\boldsymbol{\alpha}_{1:H,0}(1) \leftarrow 1.0$

$\boldsymbol{\alpha}_{1:H,0}(n) \leftarrow 0.0, n = 2, ..., N$

$\boldsymbol{u}_{1:H,0} \leftarrow 0.5$

for t=1 to T do:

$\boldsymbol{\alpha}'_{1:H,t}(n) \leftarrow ((1 - \boldsymbol{u}_{1:H,t-1})\boldsymbol{\alpha}_{1:H,t-1}(n)$
$+\boldsymbol{u}_{1:H,t-1}\boldsymbol{\alpha}_{1:H,t-1}(n-1))Attn_{1:H,t}(n))$

$\boldsymbol{\alpha}_{1:H,t}(n) \leftarrow \boldsymbol{\alpha}'_{1:H,t}(n) / \sum_{i=1:N} \boldsymbol{\alpha}'_{1:H,t}(i)$

$\boldsymbol{c}_{1:H,t} \leftarrow \sum_{i=1:N} \boldsymbol{\alpha}_{1:H,t}(i) head^{\boldsymbol{V}}_{1:H,i}$

$\boldsymbol{u}_{1:H,t} \leftarrow DNN(\boldsymbol{c}_{1:H,t}, \boldsymbol{o}_{1:H,t-1}, head^{\boldsymbol{Q}}_{1:H,t})$

end for

표 3.1: Multihead forward attention with transition agent

인코더-디코더 어텐션에서 포워드 어텐션 알고리즘은 중첩된 디코더 블럭 중 한 블럭에서만 사용하였다. 이는 트랜스포머 기반 음성합성에서 인코더 출력과 디코더 재귀 어텐션의 출력의 단조증가적인 정렬이 여러 블럭 중 한 블럭에서만 나타나며, 나머지 블럭에서는 텍스트와 멜 스펙트로그램의 정렬과는 무관한 형태의 어텐션 가중치 분포가 나타나기 때문이다. 한 블럭만의 포워드 어텐션을 통해서도 단조증가 형태의 정렬이 이루어지며, 음성합성 시 성공적으로 정렬된 음성이 생성되는 것을 확인할 수 있었다. 학습이 이루어진 후 인코더-디코더 어텐션의 어텐션 정렬은 다음과 같다.

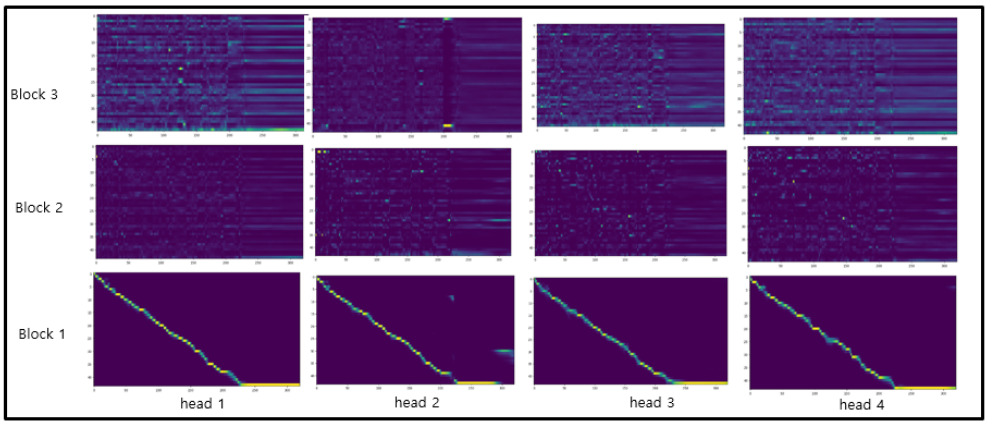

그림 3.2: Attention plot in encoder-decoder attention

3.1.4 PostNet

디코더에서 생성된 출력은 다음 단계의 멜 스펙트로그램과 스탑 토큰을 추정하는 신경망의 입력으로 사용된다. 멜 스펙트로그램은 256차의 선형 신경망과 5의 커널 크기를 갖는 5개의 256차 컨벌루션 신경망을 사용하고 선형 신경망의 출력과의 잔여 네트워크를 거쳐 추정되며, 정답 멜 스펙트로그램과의 L1 손실을 사용하며, 스탑 토큰은 256차의 선형 신경망을 거쳐 발화의 종점인지 여부를 결정하는 1차 출력을 추정하며 이진 크로스 엔트로피 손실을 사용하여 학습을 진행하게 된다. 다음 스텝의 스펙트로그램을 계산할 때는 현재 스텝까지의 스펙트로그램을 사용하여 시간에 대해 인과적으로 계산하며, 스탑 토큰은 실제 음성 생성시 스탑 토큰을 매 스텝에서 계산하여 일정 확률 이상 값으로 나타날 경우 샘플 생성을 중단하고 스펙트로그램 추정을 마치게 된다.

제 4 장 Experiments

4.1 Experimental setup

본 논문의 실험은 LJSpeech 데이터셋[15]을 사용하여 학습을 수행하였다. 데이터셋은 약 24시간 분량의 여성화자 영어 데이터로 13,100 문장으로 구성되어 있다. 멜 스펙트로그램은 80차이며, 주파수 도메인의 신호처리에서 12.5밀리초의 프레임 단위로 윈도우 크기는 50밀리초의 구성을 사용하였다. 멜 스펙트로그램은 위치-민감성 해싱 어텐션을 사용하기 위해 버킷 크기의 2배의 배수로 0-패딩을 수행하였다. 멜 스펙트로그램에서 음성 파형으로 복원하는 보코더는 WaveGlow 모델을 사용하였으며, 정답 멜 스펙트로그램 시퀀스로부터 학습한 WaveGlow 모델을 사용하였다. 학습에 사용한 GPU는 1개의 TITAN RTX로 GPU에 캐시 가능한 최대 메모리를 사용하여 학습을 진행하였다. 학습에 사용한 옵티마이저는 Adam 옵티마이저 [16]로 β_1 = 0.9, β_2 = 0.999, $epsilon$ = 10^{-8}로 옵티마이저를 구성하였고, 0.0005의 고정된 러닝레이트를 사용하였다.

4.2 Evaluation

일정한 길이의 시퀀스에 대해서 메모리를 측정하고자 텍스트와 멜 스펙트로그램을 각각 256과 1024로 설정하였다. 리포머 네트워크를 사용한 음성합성 모델과 트랜스포머 기반 음성합성 모델을 학습하는 과정에서 최대로 캐시된 메모리는 다음과 같다.

The number of memory cached			
Model	Batch size	Number of parameters	Memory cached
Transformer	32	12.47×10^6	16.76×10^9
Reformer	32	12.37×10^6	10.59×10^9
Transformer	44	12.47×10^6	24.17×10^9
Reformer	84	12.37×10^6	24.20×10^9

표 4.1: Comparison of cached memory consuming on Transformer and Reformer TTS

표 4.1에서 리포머 네트워크를 활용한 음성합성을 사용하였을 때, 트랜스포머 네트워크를 사용한 음성합성 모델에 비해 학습 과정에서 거의 절반의 메모리를 사용하여 학습이 가능한 것을 확인할 수 있었다. 만일 메모리를 크게 확보할 수 없는 환경에서 음성합성 모델을 훈련시키고 싶다면 리포머 네트워크를 사용하여 효율적인 메모리 사용에 기반한 학습이 가능할 것이다. 또한, 충분한 크기의 메모리를 사용할 수 있더라도, 음성합성 모델의 파라미터를 크게 확보하여 표현력을 높일 수 있고, 학습 과정에서 배치 사이즈를 더 크게 확보하여 안정적인 학습 또한 가능할 수 있다.

한편, 트랜스포머와 리포머 네트워크 기반의 음성합성은 실제 음성 생성 과정에서 스펙트로그램을 순차적으로 추정하게 된다. 음성 생성 과정에서 사용되는 입력은 현재까지의 스펙트로그램 전체를 사용하게 되며, 재귀 어텐션에서 매 샘플의 추정 시마다 이전까지의 샘플을 모두 사용하게 된다. 이 경우 메모리 복잡도는 $O(N^2)$이며 트랜스포머 음성합성에서는 25초 미만의 발화에 해당하는 약 2000 샘플의 생성 과정에서 계산에 실패하였다. 반면, 리포머 기반의 음성합성에서는 전체 샘플을 버켓팅한 후 버켓 크기끼리의 고정된 어텐션을 수행하기 때문에, 메모리 복잡도는 $O(NB)$가 되어 제곱에 비례하지 않으며 두 배 이상의 길이의 합성이 가능하였다.

본 논문에서 제안하는 음성합성 모델의 음질 성능을 시험하기 위해, 110개의 테스트셋 텍스트를 선별하여 트랜스포머 기반 음성합성 모델과 제안하는 모델을

통해 음성을 생성하였다. 17명의 시험자를 통해 동일한 텍스트를 발화하는 문장에 대하여 CMOS (Comparative Mean Opinion Score)를 수행하였다. 어떤 음성이 어느 모델인지 순서를 알 수 없도록 듣는 순서는 무작위로 섞은 상태에서 테스트를 진행하였다. 시험자들은 청취한 샘플에 대해 다섯 가지 분류(A가 B에 비해 매우 좋음, 좋음, 보통, 나쁨, 매우 나쁨)로 선호도를 평가하였으며, 평가는 각각 -2점부터 2점까지의 정수로 환산되었다. 또한, 동일한 테스트 음성 샘플로부터 Google API의 음성인식 서비스를 사용하여 WER (Word Error Rate)을 측정하였다. WER 평가 지표를 계산하는 식은 아래와 같다.

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C} \tag{4.1}$$

(4.1)의 식에서 S는 대체 단어 수, D는 생략 단어 수, I는 신규 단어 수, C는 일치 단어 수, N은 기준 단어 수로 $N = S + D + C$이다. WER은 음성인식 시스템에서 입력 음성에 대해 단어 단위에서 얼마나 성공적으로 음성인식을 수행하였는지 평가하는 지표이며, 음성합성 분야에서는 발화의 정확도 및 명료도를 평가하기 위해 이미 학습된 음성인식 시스템에 합성된 음성을 입력으로 사용하여 평가 지표로 사용하는 방식이 채용된 바 있다[17].

CMOS와 WER의 성능 평가 결과는 다음과 같다.

Model	CMOS	WER
Transformer with batch size 44	-	18.5%
Reformer with batch size 84	-0.035	17.7%

표 4.2: CMOS test and WER on Transformer and Reformer TTS

트랜스포머 네트워크를 사용한 음성합성 모델이 리포머 네트워크를 사용한 모델에 비해 선호도 측면에서 조금 더 나은 결과를 보였으나 점수는 전체 샘플 중 30 샘플당 1점 정도의 차이로 그 차이가 미미하였고, WER에서는 오히려 리포머 네트워크를 사용한 음성합성 모델이 더 좋은 성능을 기록하였다. CMOS에 비하여 WER 성능이 높게 기록될 수 있었던 이유로는 멀티 헤드 포워드 어텐션을 이용하여 텍스트와 스펙트로그램간의 정렬이 안정적으로 이루어졌기 때문에 발화의 이해도 및 명료도가 명확하게 드러났고, 트랜스포머 네트워크 기반 음성 합성 모델에서는 텍스트의 스크립트가 주어지지 않고 음질 테스트를 수행하기 때문에 문장 내 생략된 단어를 시험자들이 인지하지 못했기 때문이다. 이를 통해 메모리 면에서 효율적인 어텐션 구조와 잔여 네트워크 구조를 사용하더라도 음성합성의 표현력을 크게 손상시키지 않는 것을 확인할 수 있었다.

제 5 장 Conclusion and discussion

본 논문에서는 리포머 네트워크 기반의 음성합성 모델을 제안하였다. 제안하는 모델의 성능을 고성능의 신경망 음성합성 모델과 비교하였을 때, 높은 성능의 음성합성 모델을 훈련시키는 것이 비교적 저비용의 신경망 학습 [**?**]환경에서도 가능함을 확인할 수 있었다. 본 논문에서 제안하는 모델은 메모리에서 효율적인 학습이 가능하지만, 위치-민감성 해싱을 사용할 때 해시 함수 적용 후 정렬 과정이 동반되어 학습 속도가 다소 느려지는 점을 확인하였는데, 정렬 과정을 생략할 수 있는 해시 함수를 고안하여 시간 복잡도가 $O(1)$이 가능한 위치 민감성 해싱 어텐션을 사용할 경우 더욱 빠른 학습이 가능할 것이며, 이와 관련하여 속도 측면에서 트레이드-오프를 최소화하는 알고리즘을 연구할 예정이다.

한편, 재귀 어텐션에서는 위치-민감성 해싱 어텐션을 이용하면서 점연산 기반의 어텐션을 수행하기 때문에 병렬 연산을 지원하는 파이썬과 같은 인터프리터 (interpreter)에서 동시에 계산이 가능하여 합성과정에서 샘플 수가 누적이 되더라도 빠른 계산이 가능하다. 하지만 인코더-디코더 어텐션은 시퀀스 샘플에 대해 순차적으로 어텐션 계산을 수행하게 되어 샘플 수가 누적될수록 샘플 수에 비례하여 계산 시간이 늘어나게 된다. 이를 해결하기 위해서 병렬 연산이 가능한 포워드 어텐션 알고리즘을 고안할 수 있다면 보다 빠른 음성합성이 가능할 것이다.

참고 문헌

[1] A. W. Black, H. Zen, and K. Tokuda, "Statistical parametric speech synthesis," *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4.IEEE, 2007, pp. IV–1229.

[2] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 3. IEEE, 2000, pp. 1315–1318.

[3] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raimanet al., "Deep voice: Real-time neural text-to-speech," *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 195–204.

[4] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[5] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006-4010, 2017.

[6] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan et al., "Natural tts synthesis by conditioning wavenet on

mel spectrogram predictions," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4779-4783.

[7] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 33, 2019, pp. 6706–6713.

[8] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4789–4793.

[9] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4784–4788.

[10] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *International Conference on Learning Representations,* 2019.

[11] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, and L. Schmidt, "Practical and optimal lsh for angular distance," in *Advances in neural information processing systems*, 2015, pp.1225–1233.

[12] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," in *Advances in neural information processing systems,* 2017, pp. 2214–2224.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2016, pp. 770–778.

[14] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[15] K. Ito, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014

[17] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," *Neural Information Processing Systems*, 2019, pp. 3165–3174.

ABSTRACT

최근 음성합성 시스템은 신경망 기반의 종단형 음성합성 모델이 좋은 성능을 보이고 있다. 특히, 어텐션 메커니즘 기반의 시퀀스-투-시퀀스 모델은 텍스트와 스펙트로그램의 정렬과 함께 성공적으로 음향 모델링을 해내고 있다. 또한, 트랜스포머 모델 기반의 음성합성 모델은 사람의 목소리에 가까운 음성신호를 만들수 있다고 보고되었다. 하지만, 이러한 시퀀스-투-시퀀스 모델들은 많은 메모리 소모와 계산량을 요구되는데, 어텐션 에너지 값이 매 쿼리 시퀀스에 대해 키 시퀀스 전체에 대해 계산을 수행하기 때문이다. 이 문제를 해소하기 위해, 본 논문에서는 리포머 네트워크 기반 음성합성을 제안한다. 리포머 네트워크는 위치-민감성 해싱과 가역 잔여 네트워크를 사용하여 트랜스포머에 비해 메모리를 효율적으로 사용하여 모델을 학습할 수 있다. 본 논문에서는 실험을 통해 리포머 네트워크가 트랜스포머 네트워크에 비해 거의 절반의 메모리를 사용하여 음성합성 모델을 훈련할 수 있는 것을 확인하였다. 실험을 평가하기 위해 메모리 사용과 객관적, 주관적 성능평가를 사용하였다.

주요어: 음성합성, 어텐션 기반 종단형 음성합성, 리포머 네트워크

학번: 2018-23443

감사의 글

휴먼인터페이스 연구실에 들어오고 2년 6개월이 지나 곧 석사 졸업을 앞두고 있습니다. 처음 연구실에 들어왔을 때의 저를 생각해보니 연구원들 덕분에 지금까지 정말 많은 것을 배우고 익힐 수 있었다고 생각됩니다. 그동안의 시간에 많은 도움을 주신 분들에게 감사 인사를 드리고 싶습니다. 먼저, 연구실 입학을 허락해주시고 많은 가르침을 주신 김남수 교수님께 진심으로 감사의 인사를 드리고 싶습니다. 랩 미팅에서든 수업에서든 연구 지도를 받을 때마다 교수님께 큰 가르침을 받았던 기억이 있습니다. 연구 지도를 주실 때 외에도 연구원들을 생각해주시고 많은 배려를 해주실 때마다 그런 교수님을 본받고 싶다는 생각을 하였습니다. 비록 저는 석사 졸업으로 마치게 되지만, 박사 과정까지의 뜻이 있다면 김남수 교수님께 지도를 받는 것보다 좋은 연구 환경을 찾기는 쉽지 않을 것이라고 생각합니다. 항상